

ShellExecute

Use fully qualified executable filename. Do not depend on the shell's heuristics to locate the file.

Sean Barnum, Cigital, Inc. [vita¹]

Copyright © 2007 Cigital, Inc.

2007-04-16

Part "Original Cigital Coding Rule in XML"

Mime-type: text/xml, size: 6469 bytes

Attack Category	<ul style="list-style-type: none">• Path spoofing or confusion problem	
Vulnerability Category	<ul style="list-style-type: none">• Indeterminate File/Path• Process management	
Software Context	<ul style="list-style-type: none">• Shell Functions• Process Management	
Location	<ul style="list-style-type: none">• shellapi.h	
Description	<p>When using ShellExecute, ensure that the correct program is found.</p> <p>The ShellExecute() function executes a file or object. The executable filename should be fully qualified, including the file extension. Make sure you provide an unambiguous definition of the application that is to be executed. Do not depend on the shell's heuristics to locate the file.</p> <p>If any elements of the command line string contain white space, wrap them in double quotes. Otherwise, the parser might interpret an element containing one or more spaces as multiple separate elements.</p> <p>The Windows APIs ShellExecuteA, ShellExecuteW, ShellExecuteEx, ShellExecuteExA, and ShellExecuteExW are synonymous with ShellExecute.</p>	
APIs	Function Name	Comments
	ShellExecute	
	ShellExecuteA	
	ShellExecuteW	
	ShellExecuteEx	src: lpExecInfo.lpFile parameter.
	ShellExecuteExA	src: lpExecInfo.lpFile parameter.
	ShellExecuteExW	src: lpExecInfo.lpFile parameter.

1. <http://buildsecurityin.us-cert.gov/bsi-rules/35-BSI.html> (Barnum, Sean)

Method of Attack	ShellExecute will use the search path if the full path to the program is not explicitly named. An attacker could inject a Trojan horse executable into the system by placing a "tainted" executable in a location in the search path that is found before the intended executable.		
Exception Criteria			
Solutions	Solution Applicability	Solution Description	Solution Efficacy
	When ShellExecute is called.	<p>The lpFile parameter should specify a fully qualified path. If any elements of the command line string, lpParameters, contain white space, wrap them in double quotes.</p> <p>Also, the lpDirectory parameter should be fully qualified to prevent any spooling to a relative path.</p> <p>With the "Ex" functions, The lpFile member within the SHELLEXECUTEINFO parameter block should specify a fully qualified path. If any elements of the command line string, lpParameters within SHELLEXECUTEINFO, contain white space, wrap them in double quotes.</p>	Effective at preventing spoofing based on manipulation of shell path search heuristics.
Signature Details	HINSTANCE ShellExecute(

	HWND hwnd, LPCTSTR lpOperation, LPCTSTR lpFile, LPCTSTR lpParameters, LPCTSTR lpDirectory, INT nShowCmd); BOOL ShellExecuteEx(LPSHELLEXECUTEINFO lpExecInfo);				
Examples of Incorrect Code	<pre>/* This example relies on shell's heuristics to find program */ TCHAR path[] = TEXT("MyProgram"); ShellExecute(handle, "open", path, NULL, NULL, SW_SHOWNORMAL);</pre>				
Examples of Corrected Code	<pre>/* This example relies provides an absolute path to program */ TCHAR path[] = TEXT("C:\\Programs \\MyApps\\MyProgram.exe"); ShellExecute(handle, "open", path, NULL, NULL, SW_SHOWNORMAL);</pre>				
Source References	<ul style="list-style-type: none"> • Rough Auditing Tool for Security (RATS)² • http://msdn.microsoft.com/library/default.asp?url=/library/en-us/shellcc/platform/Shell/programmersguide/sec_shell.asp³ • http://msdn.microsoft.com/library/default.asp?url=/library/en-us/shellcc/platform/shell/reference/functions/shellexecute.asp⁴ • http://msdn.microsoft.com/library/default.asp?url=/library/en-us/shellcc/platform/shell/reference/functions/shellexecuteex.asp⁵ 				
Recommended Resources	<ul style="list-style-type: none"> • MSDN Security Considerations: Microsoft Windows Shell⁶ • MSDN ShellExecute reference⁷ 				
Discriminant Set	<table> <tr> <td>Operating System</td><td> <ul style="list-style-type: none"> • Windows </td></tr> <tr> <td>Languages</td><td> <ul style="list-style-type: none"> • C • C++ </td></tr> </table>	Operating System	<ul style="list-style-type: none"> • Windows 	Languages	<ul style="list-style-type: none"> • C • C++
Operating System	<ul style="list-style-type: none"> • Windows 				
Languages	<ul style="list-style-type: none"> • C • C++ 				

Cigital, Inc. Copyright

Copyright © Cigital, Inc. 2005-2007. Cigital retains copyrights to this material.

Permission to reproduce this document and to prepare derivative works from this document for internal use is granted, provided the copyright and “No Warranty” statements are included with all reproductions and derivative works.

For information regarding external or commercial use of copyrighted materials owned by Cigital, including information about “Fair Use,” contact Cigital at copyright@cigital.com¹.

The Build Security In (BSI) portal is sponsored by the U.S. Department of Homeland Security (DHS), National Cyber Security Division. The Software Engineering Institute (SEI) develops and operates BSI. DHS funding supports the publishing of all site content.

1. <mailto:copyright@cigital.com>